

## CLAIMS

What is claimed is:

1. A method, comprising:

collecting a plurality of module-to-module interfaces from a plurality of firmware modules, wherein a module-to-module interface allows a first firmware module of the plurality of firmware modules to invoke a second firmware module of the plurality of firmware modules;

collecting a plurality of dependency expressions corresponding to the plurality of firmware modules, wherein each dependency expression of a firmware module describes the module-to-module interfaces needed for execution of the firmware module; and

sorting the plurality of firmware modules into an optimized order based on the plurality of dependency expressions and the plurality of module-to-module interfaces.

2. The method of claim 1, further comprising computing a directed acycyle graph (DAG) based on the plurality of dependency expressions and the plurality of module-to-module interfaces, the DAG to be used in sorting the plurality of firmware modules.

3. The method of claim 1 wherein collecting the plurality of module-to-module interfaces comprises collecting metadata from each module-to-module interface of the plurality of module-to-module interfaces, the metadata describing the module-to-module interfaces produced by the firmware module.
4. The method of claim 1, further comprising generating an error if a firmware module of the plurality of firmware modules includes a dependency expression that refers to a module-to-module interface that is not produced by the plurality of firmware modules.
5. The method of claim 1 wherein the plurality of firmware modules comprise a plurality of Pre-EFI (Extensible Firmware Interface) Initialization Modules (PEIMs).
6. The method of claim 5 wherein the plurality of module-to-module interfaces comprise a plurality of PEIM-to-PEIM Interfaces (PPIs).
7. The method of claim 1, further comprising generating a firmware volume (FV) for the computer system, the FV including the sorted plurality of firmware modules.
8. The method of claim 7, further comprising storing the FV in a non-volatile storage device of the computer system.

9. The method of claim 7 wherein the FV to operate in accordance with an Extensible Firmware Interface (EFI) specification.

10. The method of claim 7, further comprising generating an updated FV using an FV update utility.

11. An article of manufacture comprising:

a machine-accessible medium including a plurality of instructions which when executed perform operations comprising:

starting a firmware volume build tool to generate a firmware volume (FV) for a computer system;

collecting a dependency expression from each of a plurality of firmware modules;

collecting metadata associated with each of the plurality of firmware modules, the metadata describing the module-to-module interfaces produced by each of the plurality of firmware modules;

sorting the plurality of firmware modules into an optimized order based on the dependency expressions and the metadata; and

generating the FV, wherein the FV includes the plurality of firmware modules sorted in the optimized order.

12. The article of manufacture of claim 11 wherein execution of the plurality of instructions further perform operations comprising computing a directed acycle

graph (DAG) based on the dependency expressions and the metadata, the DAG to be used in sorting the plurality of firmware modules.

13. The article of manufacture of claim 11 wherein execution of the plurality of instructions further perform operations comprising storing the FV in a non-volatile storage device of the computer system.

14. The article of manufacture of claim 11 wherein execution of the plurality of instructions further perform operations comprising generating an error signal if the module-to-module interface of a dependency expression is not described in the metadata associated with each firmware module of the plurality of firmware modules.

15. The article of manufacture of claim 11 wherein execution of the plurality of instructions further perform operations comprising removing the metadata from each firmware module of the plurality of firmware modules.

16. The article of manufacture of claim 11 wherein the FV to operate in accordance with an Extensible Firmware Interface (EFI) specification.

17. The article of manufacture of claim 11 wherein the plurality of firmware modules includes a plurality of Pre-EFI (Extensible Firmware Interface) Initialization Modules (PEIMs) and the module-to-module interfaces include PEIM-to-PEIM Interfaces (PPIs).

18. A computer system, comprising:

a processor; and

a magnetic storage device operatively coupled to the processor, the magnetic storage device including instructions which when executed by the processor perform operations comprising:

collecting a dependency expression from each of a plurality of firmware modules;

collecting a metadata from each of the plurality of firmware modules, the metadata describing module-to-module interfaces produced by a firmware module of the plurality of firmware modules;

sorting the plurality of firmware modules into an optimized order based on the dependency expressions and the metadata; and

generating a firmware volume (FV) that includes the plurality of firmware modules sorted in the optimized order.

19. The computer system of claim 18, further comprising a non-volatile storage device operatively coupled to the processor to store the FV.

20. The computer system of claim 18, further comprising a network interface operatively coupled to the processor to receive at least one firmware module of the plurality of firmware modules.

21. The computer system of claim 18 wherein the plurality of firmware modules includes a plurality of Pre-EFI (Extensible Firmware Interface) Initialization Modules (PEIMs) and the module-to-module interfaces include PEIM-to-PEIM Interfaces (PPIs).

22. A system, comprising:

data which encodes a set of firmware modules in a predetermined order, the predetermined order defined according to:

a dependency expression associated with each firmware module of the set of firmware modules; and

metadata associated with each firmware module, the metadata describing module-to-module interfaces produced by each firmware module; and

code which executes the set of firmware modules according to the predetermined order.

23. The system of claim 22 wherein the data which encodes the set of firmware modules includes a firmware volume.

24. The system of claim 22 wherein the code is executed during a pre-boot phase of a computer system.

25. The system of claim 22 wherein the code which executes the set of firmware modules includes a PEI foundation module.

26. The system of claim 22 wherein the set of firmware modules includes a plurality of Pre-EFI (Extensible Firmware Interface) Initialization Modules (PEIMs) and the module-to-module interfaces include PEIM-to-PEIM Interfaces (PPIs).

27. The system of claim 22 wherein the data and the code substantially comply with an Extensible Firmware Interface (EFI) specification.